第3章

高速シリアル通信を行う際のSerDes機能の実現法

外付けSerDesチップ vs. FPGA内蔵SerDesマクロ

皆川 翔

差動インターフェースは，パラレル・バスにも適用できるが，LSI間やボード間の高速シリアル伝送で使われることも多い．しかしLSIの内部では，データはパラレルに処理されている．そこで重要になるのがSerDes（シリアライザ/デシリアライザ）である．ここでは，LSI間やボード間で高速シリアル通信を行う際のSerDes機能の実現方法を考える．（編集部）

さまざまな電子機器で大容量データを扱うようになりました．データの転送レートを上げる方法の一つに，バス幅の拡張があります．しかしこの方法は，プリント基板の高密度化，および配線を通るデータの高速化に伴う信号の反射やクロストーク・ノイズなどが問題となる場合があります．また，機器の小型化に伴う配線の物理的な制約の問題もあります．

プリント基板上の配線を減らさなければならないときやボード間をつなぐ配線数に制約があるとき，解決策の一つとして，パラレル・データをシリアルに変換して送受信する方法があります．シリアル伝送であれば，配線数は少なくてすみます．

パラレル・データをシリアル・データに変換したり，シリアル・データをパラレル・データに戻す際には，SerDes（シリアライザ/デシリアライザ）と呼ばれる機能を使います．また，信号を高速にやりとりする必要があるため，差動インターフェースが用いられます．SerDesには，シリアル-パラレル変換のほか，高速シリアル伝送で使われる多くの機能が統合されています．

本稿では典型的なシステムにおける高速シリアル・インターフェースの適用法を解説します．とくに，設計条件に見合ったSerDes機能の実現方法に注目します．

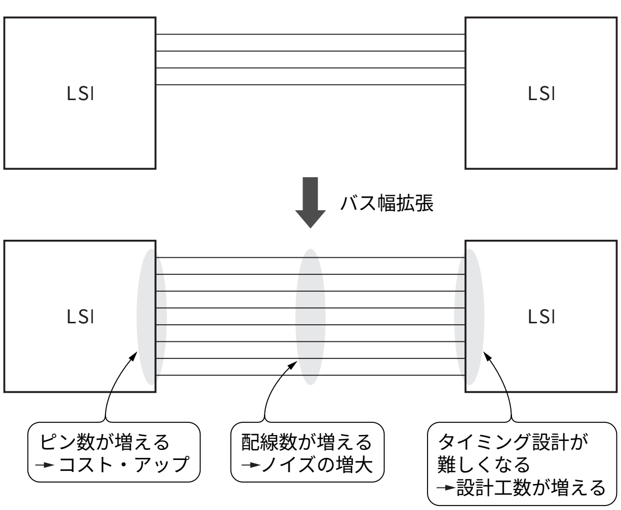


図1　高速パラレル・データ伝送の問題

データ・バス幅を広げれば，大容量データ伝送が可能になるが，多くのI/Oピンが必要になり（コスト・アップ），プリント基板上の配線数が増え（ノイズの増大），タイミング設計が難しくなる（開発工数の増大）などの問題を招くことがある．

1 高速パラレル・データ伝送の問題

1枚のプリント基板に，数多くのASICやFPGAが実装されることがよくあります．そして多くの場合，これらのLSIの間でデータのやり取りが行われます．

● 多ピンのLSIはコスト・アップにつながる

LSI間で大量のデータをやり取りしたい場合，真っ先に考えつくのはデータ・バス幅を広げる方法だと思います（**図1**）．

LSIの集積度が上がるに従って，パッケージのピン数も増えています．例えば，FPGAでも900ピン以上のI/Oピンを持つ製品があります．多ピンのパッケージを選択し，LSI間を多ビット・パラレルでつなげば，大量のデータ伝

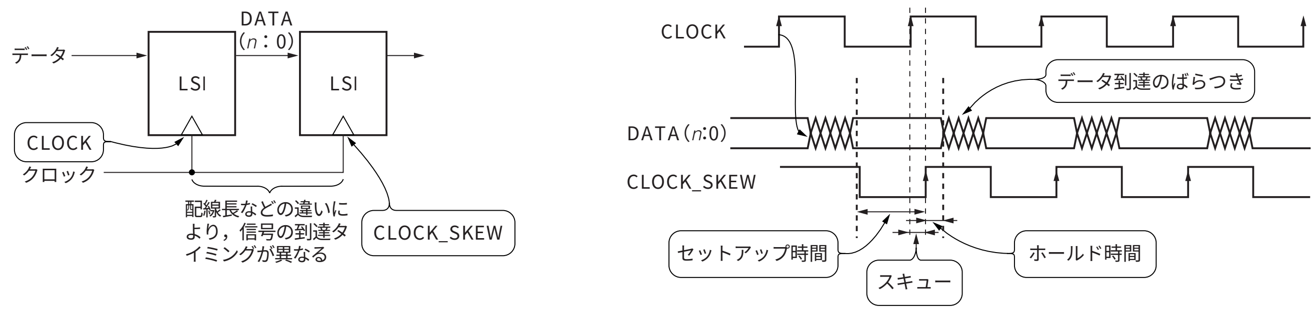


図2　高速データ伝送はタイミングが難しくなる
パラレル・データの伝送では，プリント基板上のパターンの引き回しの影響でデータ到達時間にばらつきが生じる．このため，すべてのデータが確定するタイミングが遅れ，セットアップ時間が厳しくなる．LSI間のクロックにスキューによっても，セットアップ時間やホールド時間が厳しくなる．CLOCK_SKEWはCLOCKに対して早くなることも遅くなることもありうる．

送は容易に実現できそうです．

しかし，多ピン・パッケージのLSIは，少ピン・パッケージのLSIと比べて高価になる傾向があります．LSI間のデータ伝送のために高価なLSIを使うことは，コストの面で好ましいとはいえません．

● 高速データ伝送はタイミングが難しくなる

高速なデータ伝送を行う場合，信号のタイミングを考慮しなくてはなりません．例えば，送信側のLSIからのデータ出力タイミングと，受信側のLSIの入力セットアップ/ホールド時間をすべてのビットに対して厳密に調整しなければならないのです．

図2のような回路を考えてみます．2個のLSIには，共通のクロック信号が入力されています．しかし，プリント基板の配線長の違いなどにより，クロック信号の到達時間は異なります．このタイミングのずれがクロック・スキューです．クロック・スキューは，正負両方の場合が考えられますが，受信側LSIへのクロック入力（CLOCK_SKEW）が送信側LSIへのクロック入力（CLOCK）より遅れる場合を正とします．

LSI間をつなぐパラレル・バスのDATA信号にも配線遅延があります．そしてビットごとの配線長が異なれば，ビットごとに到達タイミングがずれることになります．

ここで，回路を正常に動作させるために，以下の条件を満足しなければなりません．

サイクル時間≧送信側LSIのクロック-出力時間
＋DATA信号の配線遅延の最大
＋受信側LSIのセットアップ時間
－クロック・スキュー

ホールド時間≦送信側LSIのクロック-出力時間
＋DATA信号の配線遅延の最小
－クロック・スキュー

この制約条件を満たすためには，LSI間のプリント基板の配線パターンの設計が容易でないことがわかると思います．

● パラレル・バスはノイズの原因になる

パラレル・バスでは，プリント基板上の配線パターンが膨大な本数となる可能性があります．高速動作によるオーバシュート/アンダシュート，同時スイッチングによるグラウンド・バウンスなどのノイズが懸念され，シグナル・インテグリティ（信号波形の品質）が低下する要因となります．

シグナル・インテグリティ改善のためには，LVDSなどの差動インターフェースの利用が考えられます．しかし差動信号は，LVTTLなどのシングルエンド信号に比べて2倍の配線が必要になってしまいます．また，1チップで実現できる差動チャネル数の問題もあります．

プリント基板間のデータをバックプレーンを経由して伝送する際，データを伝送する本数がコネクタのピン数よりも多くなってしまっては物理的に通信できないので，バス幅を増やすにも限界があります（**図3**）．

2 低コストLSIにSerDesチップを付ける

従来のパラレル伝送で設計が困難な場合，高速シリアル伝送に頼るしかありません．パラレル線をシリアル線にすることにより，伝送に必要なパターンの本数を大幅に削減することができます．

ただし，大容量のデータをシリアルで伝送するわけです

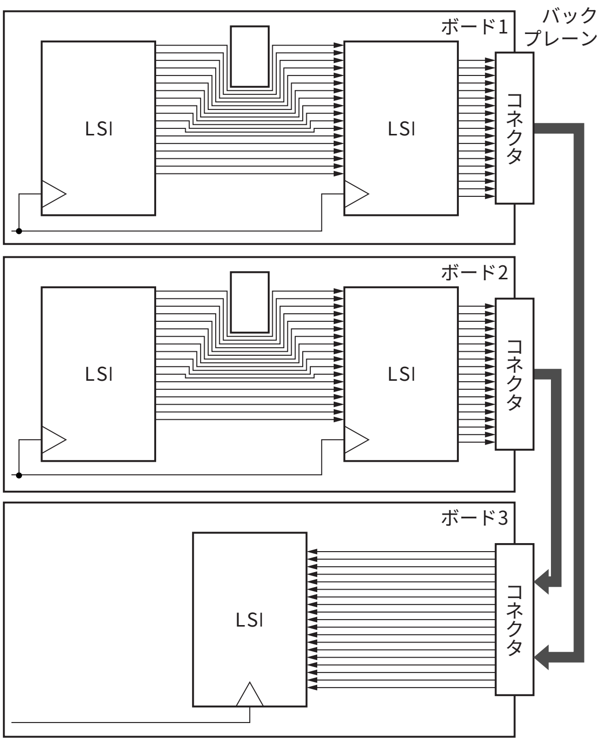


図3　パラレル伝送時の問題

LSIどうしの接続や，コネクタとバックプレーンを経由したボード間の接続の場合，さまざまな問題が発生する．高密度プリント基板の場合，LSIどうしをつなぐパラレル線が均等に配線できないことが多く，データの到達時間がビットごとに異なってしまう．高速動作時のタイミング設計が厳しくなる傾向があり，設計の難易度が高くなる．また，コネクタやバックプレーンを経由して二つのボードのデータを別のボードへ集約させる場合，集約するボードの信号線数が増える．コネクタのピン数やLSIのI/Oピン数が不足することもある．配線数の増大もノイズの原因となり，好ましくない．

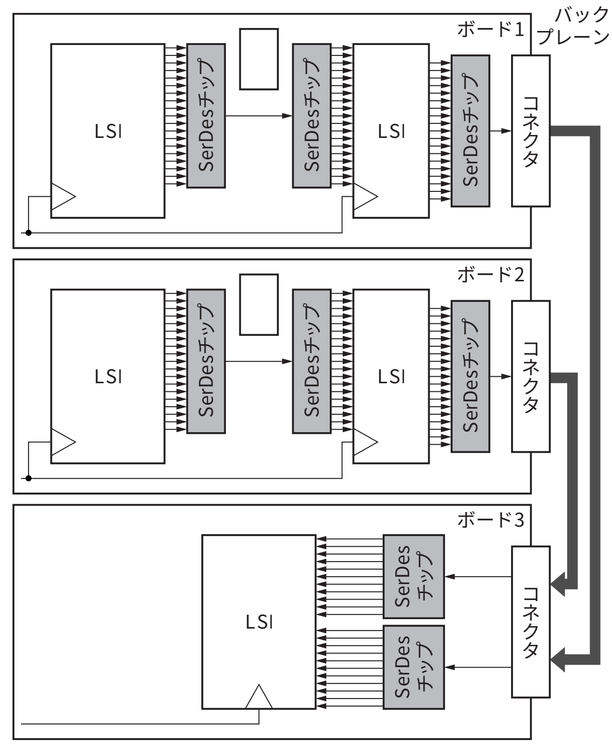


図4　SerDesチップを使用する

LSI間の通信にSerDesチップを用いることにより，プリント基板上のパターン数を減らすことができる．ボード間のデータ伝送でも，物理的制約を回避することが可能となる．ただし，LSIとSerDesチップの間はパラレル伝送になるため，タイミングに関する問題に注意して設計する必要がある．

から，データ転送速度は上がります．LSI内部では，データはパラレルに扱われることがほとんどのため，シリアライザ/デシリアライザ（SerDes）が必要になります．動作速度の確保やDCバランス注1のために符号化回路などを設計するのは多大な工数がかかりますし，検証にも時間がかかってしまいます．

● SerDesチップを使用する

高速シリアル伝送を手軽に実現する方法の一つに，SerDesチップの利用があります（**図4**）．SerDesチップを使用するメリットとしては，

- 比較的安価に購入できる
- 物理層の制御が比較的容易
- SerDesチップ間のシグナル・インテグリティを改善できる
- プリント基板の配線領域を確保しやすい

などが挙げられます．SerDesチップは高価なように見られがちですが，中には気軽に利用できる製品もあります．一例を**表1**に示します．

SerDesチップは，トランスミッタとレシーバに同じ製品（またはペアの製品）を使うのが基本です．SerDesチップとひと口に言っても，シリアライザ/デシリアライザの構造や伝送方式などに違いがあるためです．

表1に示した三つのSerDesチップはどれもアーキテクチャに違いがありますが，性能はほぼ同じです．それでは何を基準に選択すればよいのでしょうか．もちろん低コストであるに越したことはありませんが，価格については流通や購入量の違いで大きく変動するケースがあるので，ここでは比較を行いません．コスト以外で基準となるのは，

注1：‘0’と‘1’の出現頻度を均等にすること．例えば8b/10bエンコーディングを行うと，‘0’/‘1’が長く連続せず，DCバランスをとることができる．データ伝送時，‘0’や‘1’のデータが連続すると送信側と受信側のデバイス間にDC電流成分が生じる，その結果，データ・サンプリング時のマージンが少なくなるという問題が発生する．シリアル伝送では符号化処理を行うことによって‘0’や‘1’の連続したデータを抑え，DCバランスを改善することができる．

表1　SerDesチップの例

型　名	メーカ	データ転送速度(最大)	チャネル数	伝送方式	DCバランス	パラレル・ポート	ピン配置
DS90C387	米国National Semiconductor社	5.38Gbps	Tx＝8レーン	ソース・シンクロナス	対応	48ビット	固　定
DS90CF388	米国National Semiconductor社	5.38Gbps	Rx＝8レーン	ソース・シンクロナス	対応	48ビット	固　定
ispGDX2	米国Lattice Semiconductor社	3.5Gbps	Tx＝8レーン，Rx＝8レーン	ソース・シンクロナス，CDR	対応	64ビット	可　変
PM8358	米国PMC-Sierra社	12.8Gbps	Tx＝4レーン，Rx＝4レーン	CDR	対応	64ビット	固　定

やはり使いやすさだと思います．

1) DS90C387/DS90CF388

DS90C387/DS90CF388は，イネーブル・ピンを有効にすることで，DCバランス処理を行ってくれます．単純でとても扱いやすいSerDesチップだと思います．手っ取り早く使うにはお勧めです．しかし，ちょっとしたカスタマイズが必要なケースなどにはあまり向いていません．

伝送方式はソース・シンクロナス方式です．シリアルのクロック線はデータとは独立して受け渡すことになるので，データ線のほかにクロック線の配線が必要になります．

2) ispGDX2

ispGDX2は，ソース・シンクロナスとCDR(クロック・データ・リカバリ)の両方の伝送方式に対応しています．内部アーキテクチャもチャネル単位で区切られており，内部配線をカスタマイズすることが可能です．ピン配置を指定することができ，プリント基板のレイアウト条件にも柔軟に対応できるようになっています．

プログラマブルな特徴を持つということは，逆にいえば配線条件やピン配置を指定する作業がかならず発生するということです．手軽に使ってみるというわけにはいかないようです．

3) PM8358

PM8358の伝送方式はCDRのみです．データ転送速度は，通常のモードで1.2Gbps～3.2Gbpsです．4チャネルのデータ・ポートのほか，完全冗長用の追加回路が送受ともに4ポート内蔵されています．また，データ転送速度が半分になるハーフ・レート・モードを使用することによって，600Mbps～1.6Gbpsで送受各8ポートとして使用することもできます．

内部にクロス・コネクト・スイッチを内蔵しており，シリアル-パラレルのポート間割り当てをそれぞれ変更したり，1：8のマルチキャストなどを行うことも可能です．このクロス・コネクトの設定や，各種ブロックおよび機能の個別の設定は内部レジスタで行うことになり，外部デバイスからアクセスする必要があります．

ただし，基本的な動作モード設定については外部ピンからの設定のみでも対応でき，基本設定モードはスタンドアロンで使用できます．

● SerDesチップは目的によって使い分ける

手っ取り早く使いたいのであれば，DS90C387/DS90CF388がいいでしょう．プリント基板のパターン・レイアウトについてソース・シンクロナス方式では問題があるというケースなら，ispGDX2やPM8358を使用してCDRに切り替えるといいかと思います．

どのSerDesチップを使うときにも共通して注意しなくてはならないのは，あくまでもLSIから出力されるパラレル線は必要になるということです．LSIからSerDesチップまでは基板上に多くのパターンを引かなくてはなりません．LSIのI/Oピンを減らせるわけではないので，LSIとSerDesチップの間のタイミング設計には十分な注意が必要です．

3

③ FPGA内蔵のSerDes機能を使用する

プリント基板の制約のため，SerDesチップを配置することができない場合があります．SerDes機能を使うことによってバックプレーンに配線する信号線を減らせるとわかっていても，SerDesチップを使えなければ話になりません．

このような場合は，FPGAの内部に組み込まれているSerDes機能を使用するとよいでしょう(**図5**)．

● 高機能FPGAはSerDes機能を持つ

最近では，SerDes機能を持ったFPGAもあります．FPGAに内蔵されているので，SerDesチップと接続するための多くの配線も必要なくなります．SerDesチップを

表2　SerDes内蔵FPGAの例

ファミリ名	メーカ	データ転送速度	チャネル数	伝送方式	DCバランス	専用リンク層
Stratix II GX	米国Altera社	最大6.375Gbps	4～20	ソース・シンクロナス，CDR	8b/10b	SerialLite
Virtex-4	米国Xilinx社	最大10.3125Gbps	0～24	ソース・シンクロナス，CDR	8b/10b，64b/66b	Aurora

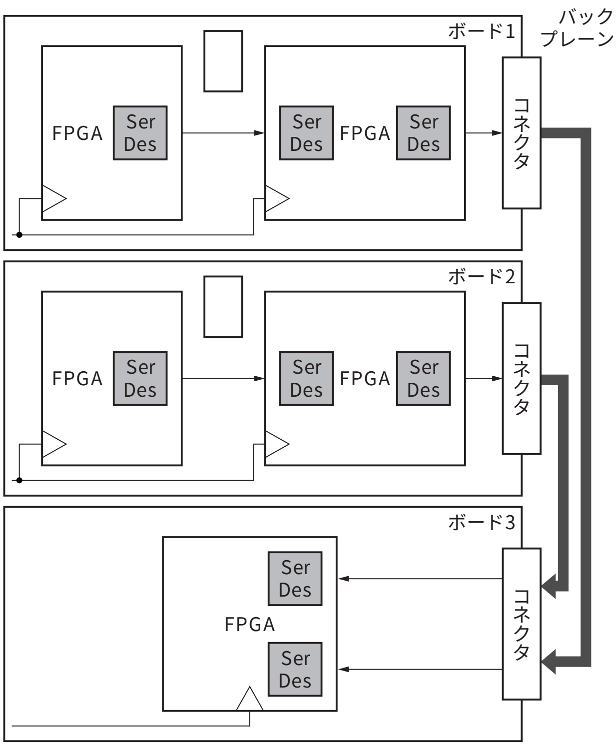


図5　FPGA内蔵のSerDes機能を使用する

FPGA内蔵のSerDes機能を用いることによって，SerDesチップとつながるパラレル・バス部分のタイミング設計の問題を解消できる．FPGA専用のリンク層もサポートされており，使いかたは容易である．しかし，SerDes機能内蔵のFPGAは高価であるため，コストが重視されるアプリケーションには向かない．FPGA内蔵のSerDes数は製品によって異なるため，注意が必要．

基板上に置かなくてすみ，また，シグナル・インテグリティが改善されることも期待できます．

ただし，FPGA内部に組み込まれているSerDes機能の数には限りがあります．また，SerDes機能内蔵のFPGAは高価なため．外付けのSerDesチップを使用するよりコスト・アップになってしまいます．

SerDes機能を持ったFPGAの例を**表2**に示します．

SerDesチップの場合と同じように，トランスミッタとレシーバについては，同じアーキテクチャのものを使ったほうがよいと思います．

データ転送速度ですが，最高性能のFPGAなら10Gbpsです．大容量のデータ伝送も可能であり，高機能のSerDesチップにもひけをとらない性能を持っています．

SerDesチップと違い，SerDes機能内蔵のFPGAはリンク層をサポートしている点が大きな特徴です． ただし，メーカごとにリンク層のアーキテクチャが異なっており，例えば米国Altera社であれば「SerialLite」，米国Xilinx社であれば「Aurora」という方式になります．リンク層を活用する前には覚えなくてはならないことがたくさんあります（右掲のコラム「Virtex-4のSerDes機能とAuroraプロトコル」を参照）．

● 高速論理回路の設計が求められる

データ転送速度が非常に高くなった場合，当然FPGA内部にあるパラレル側の速度も高速になります．このため，論理回路の高速化設計が必須です．

CPUやDSPと違い，FPGAは回路構成によって最大動作周波数が大きく変化します．フリップフロップ間で形成するランダム論理の規模が大きくなると，それだけ使用する論理ブロックが増え，その結果データ伝播遅延が増えることになります．詳細仕様を検討する段階でつねに動作速度を意識したほうがよいでしょう．フリップフロップ間のランダム論理は，冗長回路と思えるほど少量にしておいたほうが，後々問題が発生してから対策するよりはるかによいと思われます．

動作速度を引き上げる別の手段として，ツールによる対策が挙げられます．

- 配置配線の際にフロアプラン・ツールを利用してブロックの最適配置を行う．
- 配置配線ツールのオプションで，配線時の優先順位を「速度優先」に設定する．
- 論理合成時にフィジカル合成ツールを使用し，合成時の精度を高める．

ツールに頼るのはあくまでも最終手段であって，やはりその前のRTL設計時に動作速度を意識して設計するのが基本です．回路が悪ければ，ツールががんばっても大きな改善は望めません．

● 消費電力と発熱に注意

もう一つの問題は「熱」です．SerDes機能がFPGA内部に組み込まれたことによってシグナル・インテグリティは改善されました．しかし，この結果，SerDesの回路が発する熱もFPGA内部に組み込まれたことになり，パラレル伝送時よりも温度上昇率が大幅に上がることを覚悟しなくてはなりません．さらに90nmプロセス特有のリーク電流による消費電力増大も重なるため，放熱の対策はより注意深くならざるをえません．

FPGAのチップにヒートシンクを付ければ…という簡単な対策ですむ場合もあるとは思いますが，とにもかくにも詳細仕様を検討する段階で消費電力の見積もりをしっかりと行っておくべきです．詳細仕様のパラメータで消費電力を見積もるツールを各FPGAメーカが用意しているので，積極的に使用しましょう．自分が予想していた消費電力とどのくらいの誤差があるのかが把握できるだけでも，後の熱対策の検討が行いやすくなると思います．

● 目的に合わせて構成を決める

プリント基板の物理的問題，シグナル・インテグリティ問題，コストの問題など，検討事項はいろいろありますが，まずは自分が何を最優先で設計を進めるのかによって，設計のアプローチが変わってくると思います．

品質最優先ならFPGA内部のSerDes機能を使う方法がいちばん良いと思います．しかし，コストを考えると，低コストFPGAとSerDesチップを使用する組み合わせが妥当な場合もでてくることでしょう．

参考・引用*文献

(1) National Semiconductor，DS90C387/DS90CF388データ・シート．
(2) Lattice Semiconductor，ispGDX2データ・シート．
(3) Xilinx，Auror Protocolデータ・シート．

みなかわ・かける

COLUMN　Virtex-4のSerDes機能とAuroraプロトコル

Virtex-4にはソース・シンクロナス方式の「CHIP Sync」と，CDR方式で動作する「MGT（Multi Gigabit Transceiver）」と呼ばれるSerDesモジュールがハード・マクロで組み込まれています．

● シンプルなCHIP Sync

CHIP Syncのアーキテクチャは単純なシリアル-パラレル変換回路であるため，データ・レートが非常に高い場合は，DCバランス用の回路を設計する必要があります．

しかし，Chip Syncには便利な機能があります．シリアル・データを受信する際，デシリアライザの受信クロックを自動的に位相調整してくれるのです．この機能によって，受信クロックのセットアップ/ホールド時間のマージンを気にする必要がなくなります．

● 符号化機能を持つMGT

CHIP Syncでは実現が難しい，より高速なデータ・レートが必要な場合は，MGTを使用するとよいでしょう．MGTはCDR方式のため，同期クロック線は不要です．また，DCバランスのための符号化モジュールも8b/10b，64b/66bから選択が可能です．

一つのMGTモジュールには送信用のTxレーンと受信用のRxレーンが組み込まれており，これらをペアにして使用します．MGTを複数組み合わせて使用する場合は，チャネル・ボンディングという機能を使用します．この機能は，基板のパターン長や伝送ケーブルの長さの違いによって生じた受信側の位相差を自動的に修復します．

● MGTを制御するプロトコルAurora

多機能な分だけMGTによる制御用回路の設計にはかなりの知識とノウハウが必要になります．ただし，MGTを直接制御せず，リンク層を活用するという方法もあります．Xilinx社独自のリンク層として「Aurora」というプロトコルが用意されています．PCI ExpressやXAUIなどの汎用プロトコルも用意されていますが，汎用プロトコルを使う必要がないならAuroraがお勧めです．Auroraを使用することによって，ユーザがMGTの制御に関与する必要がなくなります．また，一つのリンク層で複数の物理層を使用するとき，MGTへのデータの振り分けを自動で制御してくれます．

Auroraを使用するなら，当然，Auroraの制御回路が必要になりますが，単純な送受信だけならさほど難しくはありません．エラー訂正などの機能を付ければ，さらに通信の品質を上げることができます．

● Auroraを使った送受信

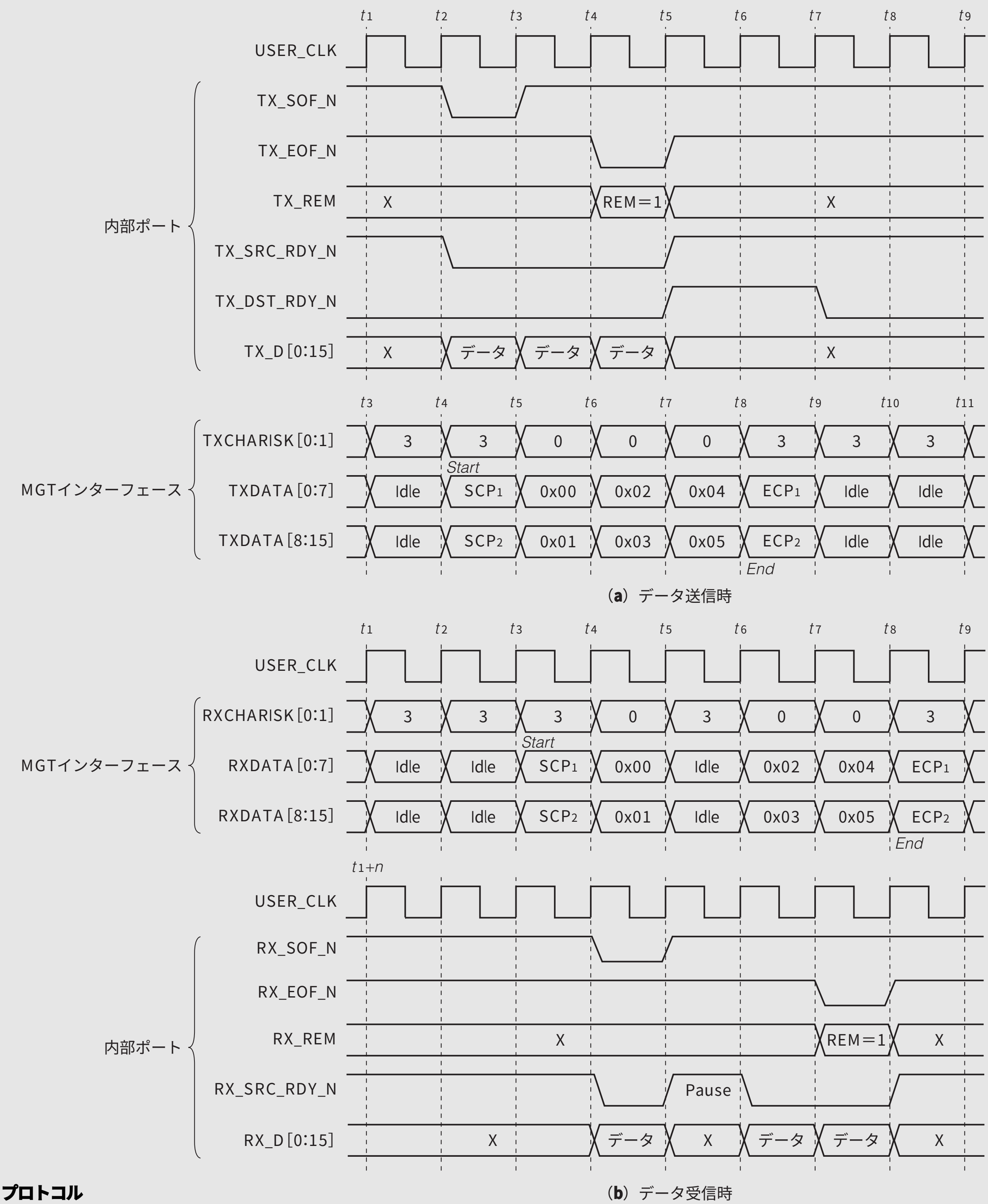
Auroraは，パケット方式でハンドシェークを行いながら通信を行います．p.122の**図A-1**にデータ送受信のタイミングを示します．

パケット送信時，先頭データと同じサイクルでTX_SOF_NとTX_SRC_RDY_Nを“L”にします．TX_SOF_Nは先頭データのサイクルのみ“L”ですが，TX_SRC_RDY_Nのデータ送信中は“L”にする必要があります．パケットの最終データ時に，TX_EOF_Nを“L”にします．同じサイクルでTX_REMを指定し，TX_SRC_RDY_Nは次のサイクルから“H”にします．

3

Auroraからへデータ転送する際，MGT内部のFIFOのアドレス・ポインタ情報がAuroraへ送られますが，この期間中TX_DST_RDY_Nが“L”になります．つまり，TX_DST_RDY_Nが“L”の間は次のパケット・データを送ることができません・パケット・データを送信する際にはTX_DST_RDY_Nが“H”になっていることを確認しなくてはなりません．

パケット・データ受信時でも，送信時と制御信号のタイミングは変わりません．パケット・データの先頭でRX_SOF_Nが1サイクル“L”，最終データにてRX_EOF_Nが1サイクル“L”になります．ただし，データの受信中，ストリームでデータを受信できるわけではないので，データの無効期間はRX_SRC_RDY_Nが“H”になり，その期間はデータを受信しないように制御しなくてはなりません．



図A-1
Auroraプロトコル